Лабораторная работа № 5

студента группы ИТз-221

Дмитриева Дмитрия Анатольевича

*Выполнение: \_\_\_\_\_\_\_\_\_ Защита: \_\_\_\_\_\_\_\_\_\_\_*

Методы класса Graphics. Элемент управления DataGridView

*Цель работы***:** знакомство с возможностями среды разработки Visual C# для работы с графикой и изображениями и получение навыков работы с графикой и изображениями при разработке приложений. Освоение приемов работы с элементом управления DataGridView для отображения данных в виде таблицы.

**Ход работы:**

***Вариант - 12***

1. Разработал приложение для построения диаграммы по введенным данным. Данные вводятся в DataGridView, диаграмма рисуется без применения элемента управления Chart, используется только экземпляр класса Graphics. Задание кольцевая диаграмма, максимальное количество входных данных – 3. График вида:



1. Используемые элементы на форме: Panel – для размещения элементов в определенном контейнере (в данном случае для отрисовки графика), DataGridView – для ввода данных графика, Button – обработчик, для построения графика. Получил следующую форму (рис. 1)

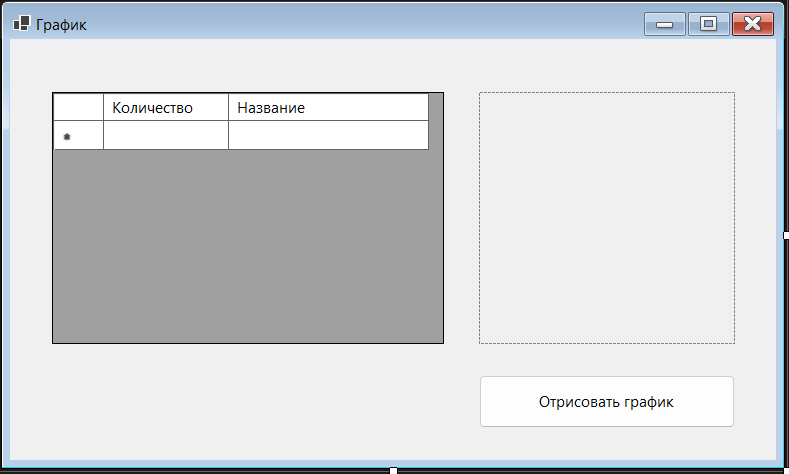


Рисунок 1 – главная форма приложения

1. Метод парсинга данных из DataGridView и вызов метода для отрисовки графика (рис. 2). В данном методе программа получает все данные из первого столбца и конвертирует их в float.

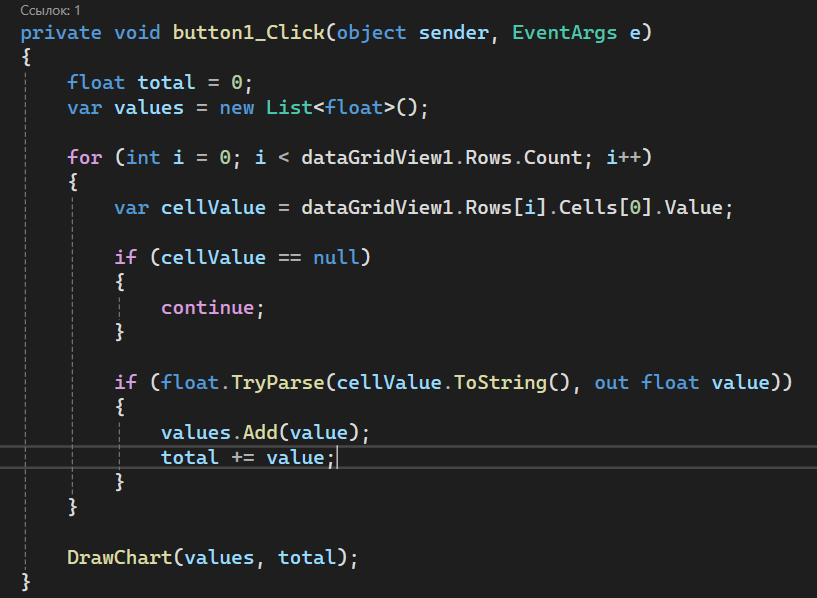
****

Рисунок 2 – метод получения данных

1. Метод отрисовки графика с использованием экземпляра класса Graphics (рис. 3). После инициализации программа очищает зону отрисовки белым цветом, вычисляются центральные координаты графика, радиус, а также кисти с узорами. После по переданным данным программа обрисовывает части круговой диаграммы, а также выполняет обводку этих частей. В конце центр диаграммы заливается белым эллипсом.

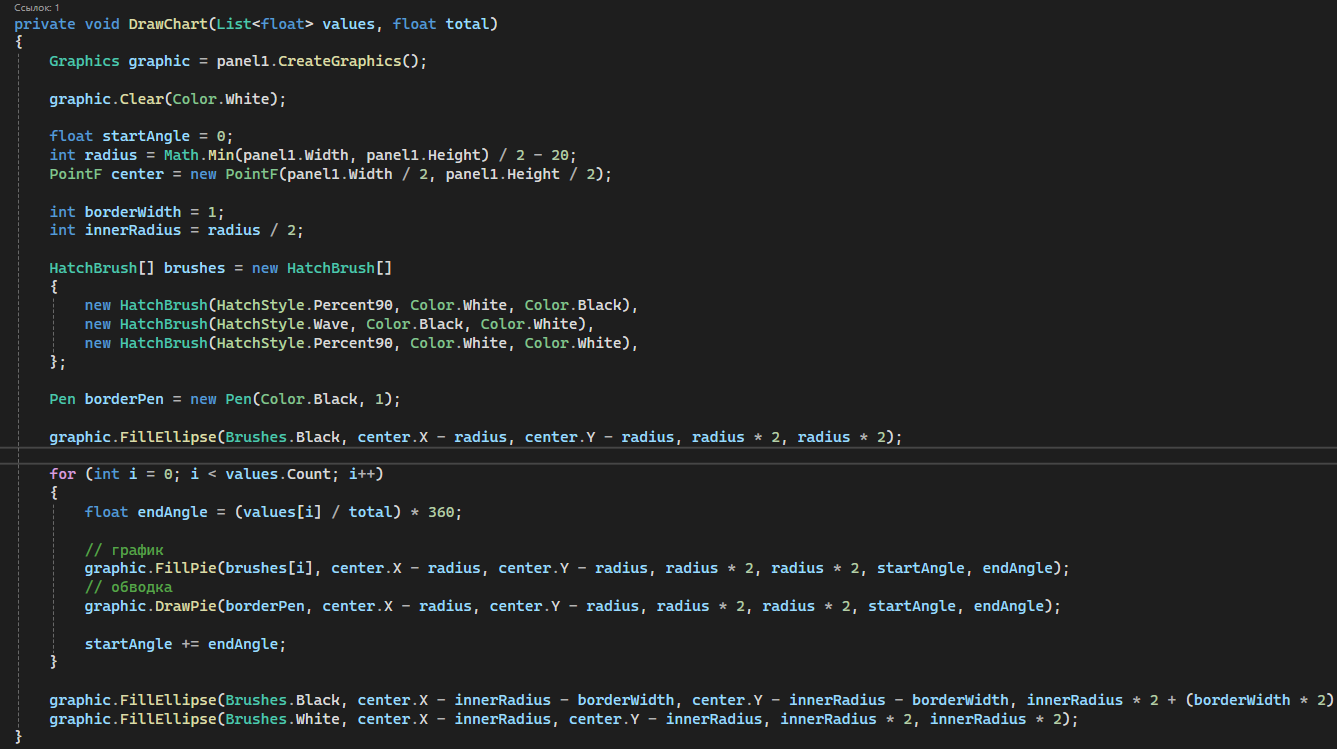
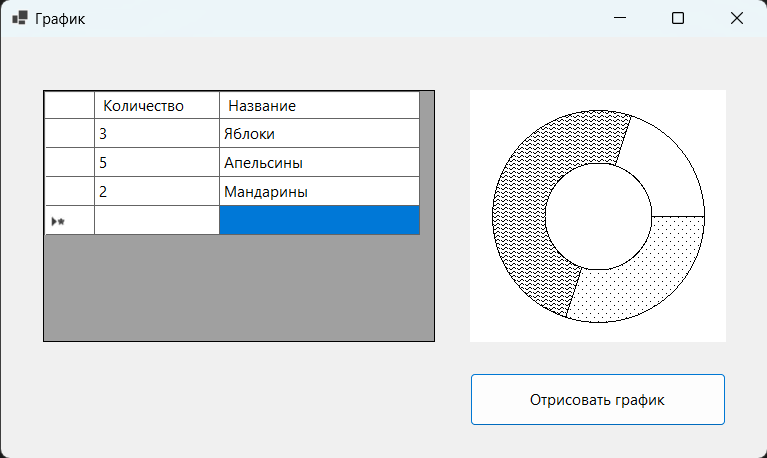


Рисунок 3 – метод отрисовки графика

1. Протестировал программу, получил два разных графика используя разные входные данные (рис. 4).



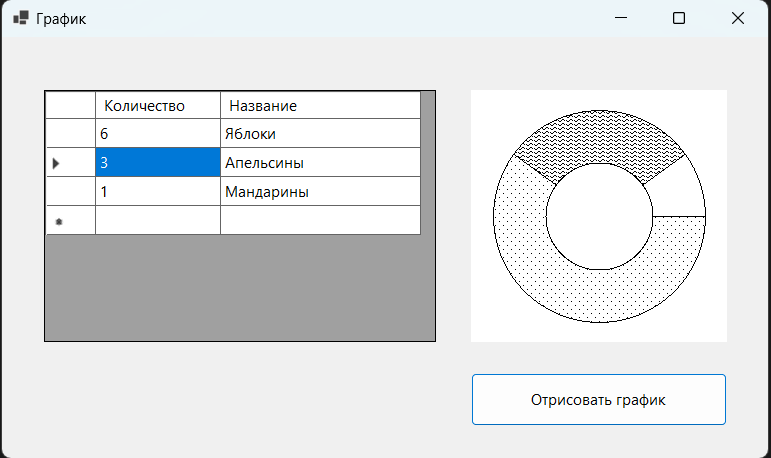


Рисунок 4 – результат выполнения программы

**Контрольные вопросы:**

1. **Интерфейс графических устройств GDI**: GDI (Graphics Device Interface) — это API для рисования и вывода графики в Windows-программах, позволяющий работать с изображениями, текстами и графическими примитивами.
2. **Методы класса Graphics**: Класс Graphics предоставляет методы для рисования, такие как DrawLine(), DrawRectangle(), DrawImage(), и FillRectangle() для рисования линий, прямоугольников, изображений и заливки.
3. **Система координат окна, для которого получен объект Graphics**: Координаты окна используют систему с началом в верхнем левом углу, где ось X направлена вправо, а ось Y — вниз.
4. **Варианты метода DrawLine для рисования одной и нескольких линий**: Метод DrawLine() рисует одну линию, а DrawLines() — несколько линий, соединяя последовательность точек.
5. **Варианты метода DrawRectangles для рисования одного и нескольких прямоугольников**: Метод DrawRectangle () рисует один прямоугольник, а DrawRectangles() — несколько прямоугольников, передавая массив прямоугольников.
6. **События MouseUp, MouseDown, MouseMove**: Эти события используются для обработки действий мыши: MouseDown — при нажатии кнопки, MouseUp — при отпускании, а MouseMove — при движении мыши.
7. **Использование объекта Bitmap**: Объект Bitmap представляет изображение в памяти и позволяет создавать, изменять и сохранять растровые изображения, а также манипулировать пикселями.
8. **Использование объекта PictureBox**: Элемент PictureBox используется для отображения изображений в формах Windows Forms, поддерживает различные форматы изображений и операции с ними.
9. **Назначение и свойства элемента управления DataGridView**: DataGridView используется для отображения данных в табличном виде, с возможностью редактирования, сортировки и фильтрации строк и столбцов.
10. **Настройка несвязанного DataGridView**: для несвязанного DataGridView данные заполняются программно через коллекции или массивы, без привязки к источнику данных.
11. **Настройка DataGridView, связанного с источником данных**: При связывании DataGridView с источником данных (например, BindingSource, DataTable) автоматически отображаются и редактируются данные.
12. **Допустимые источники данных для DataGridView**: Источниками могут быть коллекции, такие как DataTable, BindingList<T>, IEnumerable, или другие объекты, реализующие интерфейс IList или IEnumerable.
13. **Обеспечение двустороннего механизма связи между DataGridView и коллекцией**: для двусторонней связи используется привязка данных через BindingSource, что позволяет изменениям в таблице автоматически обновлять коллекцию и наоборот.

**Вывод**: ознакомился с возможностями среды разработки Visual C# для работы с графикой и изображениями и получил навыки работы с графикой и изображениями при разработке приложений. Освоил приемы работы с элементами управления DataGridView для отображения данных в виде таблицы.